What is claimed is:

1. A communication method comprising the steps of:

providing a first executable application at a server, the first executable application requiring a first amount of memory for proper execution;

executing a second executable application at a client, the second executable application occupying a second amount of memory;

providing a first data portion at a client, the first data portion occupying a third amount of memory, the first data portion being accessible by the second executable application;

providing a second data portion at a client, the second data portion having a lower priority than the first data portion, the second data portion occupying a fourth amount of memory, the second data portion being accessible by the second executable application;

receiving an input at the client, the input corresponding to the first executable application on the server; and

determining, in response to receiving the input, whether the client has sufficient available memory for execution of the first executable application.

2. The method of claim 1, wherein the step of providing the first data portion further comprises the step of:

providing a current program data file.

3. The method of claim 2, wherein the step of providing the second data portion further comprises the step of:

providing a future program data file.

4. The method of claim 1, further comprising the step of:

downloading the first executable application in response to determining that the client has sufficient available memory for execution of the first executable application.

5. The method of claim 1, further comprising the steps of:

purging the second data portion from the memory in response to determining that the client has insufficient available memory for execution of the first executable application; and

deciding, in response to purging the second data portion, whether the client has sufficient available memory for execution of the first executable application.

6. The method of claim 5, further comprising the step of:

downloading the first executable application in response to deciding that the client has sufficient available memory for execution of the first executable application.

7. The method of claim 5, further comprising the steps of:

purging the first data portion from the memory in response to further determining that the client has insufficient available memory for execution of the first executable application; and

assessing, in response to purging the first data portion, whether the client has sufficient available memory for execution of the first executable application.

8. The method of claim 7, further comprising the step of:

downloading the first executable application in response to assessing that the client has sufficient available memory for execution of the first executable application.

9. The method of claim 7, further comprising the steps of:

purging the second executable application from the memory in response to assessing that the client has insufficient available memory for execution of the first executable application; and

evaluating, in response to purging the second executable application, whether the client has sufficient available memory for execution of the first executable application.

10. The method of claim 9, further comprising the step of:

downloading the first executable application in response to evaluating that the client has sufficient available memory for execution of the first executable application.

11. A communication system comprising:

a client having a tuner, the client being capable of receiving broadcast information, the client being in bi-directional communication with a server, the server having an executable application, the executable application requiring an amount of memory for proper execution; and

a memory manager in the client, the memory manager being configured to determine whether the client has sufficient available memory for proper execution of the executable application, the server being configured to transmit the executable application in response to the memory manager determining that the client has sufficient memory for proper execution of the executable application.

12. A communication method comprising the steps of:

establishing a bi-directional communication pathway between a client and a server, the client having a tuner, the client being capable of receiving broadcast information, the server having an executable application, the executable application requiring an amount of memory for proper execution;

receiving an input at the client, the input corresponding to the executable application on the server; and

determining, in response to receiving the input, whether the client has sufficient available memory for proper execution of the executable application.

13. The method of claim 12, wherein the memory is a volatile memory.

14. The method of claim 12, further comprising the steps of:

allocating the required amount of memory from the available memory in response to determining that the client has sufficient available memory for proper execution of the executable application; and

requesting the executable application from the server upon allocating the required amount of memory.

15. The method of claim 12, further comprising the steps of:

purging data contained in the memory in response to determining that the client has insufficient available memory for proper execution of the executable application; and

determining, in response to purging the data, whether the client has sufficient available memory for proper execution of the executable application.

16. The method of claim 15, further comprising the steps of:

purging a pre-existing application contained in the memory in response to determining that the client has insufficient available memory for proper execution of the executable application after purging data contained in the memory; and

determining, in response to purging the pre-existing application, whether the client has sufficient available memory for proper execution of the executable application.

17. The method of claim 15, wherein the step of purging data comprises the step of:

purging the memory in accordance with a dynamic list of priorities.

18. The method of claim 15, wherein the step of purging the memory comprises the steps of:

purging a data file having television viewing information for future days prior to purging a data file having television viewing information for a current day; and

purging a data file having television viewing information for a current day prior to purging an application loaded upon initialization.

19. The method of claim 15, further comprising the steps of:

iteratively repeating the purging and determining steps until the client has sufficient available memory for proper execution of the executable application;

allocating the required amount of memory from the available memory in response to determining that the client has sufficient available memory for proper execution of the executable application; and

requesting the executable application from the server upon allocating the required amount of memory.

20. The method of claim 15, wherein the step of purging the data comprises the step of purging data in accordance with a dynamic list of priorities.

21. The method of claim 12, further comprising the steps of:

purging a pre-existing application contained in the memory in response to determining that the client has insufficient available memory for proper execution of the executable application; and

determining, in response to purging the pre-existing application, whether the client has sufficient available memory for proper execution of the executable application.

22. The method of claim 21, further comprising the steps of:

iteratively repeating the purging and determining steps until the client has sufficient available memory for proper execution of the executable application;

allocating the required amount of memory from the available memory in response to determining that the client has sufficient available memory for proper execution of the executable application; and

requesting the executable application from the server upon allocating the required amount of memory.

23. The method of claim 21, wherein the step of purging the pre-existing application comprises the step of purging the memory in accordance with a dynamic list of priorities.

24. The method of claim 12, further comprising the steps of:

compacting a private heap in response to determining that the client has insufficient available memory for proper execution of the executable application; and

determining, in response to compacting the private heap, whether the client has sufficient available memory for proper execution of the executable application.

25. The method of claim 12, further comprising the steps of:

compacting a system heap in response to determining that the client has insufficient available memory for proper execution of the executable application; and

determining, in response to compacting the system heap, whether the client has sufficient available memory for proper execution of the executable application.

26. A communication method comprising the steps of:

establishing a bi-directional communication pathway between a client and a server, the client having a tuner, the client being capable of receiving broadcast information, the server having an executable application, the executable application being executable using a normal amount of memory, the executable application further being executable using a reduced amount of memory;

receiving an input at the client, the input corresponding to the executable application on the server; and

determining, in response to receiving the input, whether the client has sufficient available memory for execution of the executable application using the normal amount of memory.

27. The method of claim 26, further comprising the steps of:

allocating the normal amount of memory from the available memory in response to determining that the client has sufficient available memory for execution of the executable application using the normal amount of memory; and

requesting the executable application from the server upon allocating the normal amount of memory.

28. The method of claim 26, further comprising the steps of:

determining, in response to determining that the client has insufficient available memory for execution of the executable application using the normal amount of memory, whether the client has sufficient available memory for execution of the executable application using the reduced amount of memory.

29. The method of claim 28, further comprising the steps of:

allocating the reduced amount of memory from the available memory in response to determining that the client has sufficient available memory for execution of the executable application using the reduced amount of memory; and

requesting the executable application from the server upon allocating the reduced amount of memory.

30. The method of claim 28, further comprising the steps of:

purging data contained in the memory in response to determining that the client has insufficient available memory for execution of the executable application using the reduced amount of memory; and

determining, in response to purging the data, whether the client has sufficient available memory for execution of the executable application using the reduced amount of memory.

31. The method of claim 30, further comprising the steps of:

iteratively repeating the purging and determining steps until the client has sufficient available memory for execution of the executable application using the reduced amount of memory;

allocating the reduced amount of memory from the available memory in response to determining that the client has sufficient available memory for execution of the executable application using the reduced amount of memory; and

requesting the executable application from the server upon allocating the reduced amount of memory.

32. The method of claim 30, wherein the step of purging the data comprises the step of purging data contained in memory in accordance with a dynamic list of priorities.

33. The method of claim 28, further comprising the steps of:

compacting a private heap in response to determining that the client has insufficient available memory for execution of the executable application using the reduced amount of memory; and

determining, in response to compacting the private heap, whether the client has sufficient available memory for execution of the executable application using the reduced amount of memory.

34. The method of claim 28, further comprising the steps of:

purging a pre-existing application contained in the memory in response to determining that the client has insufficient available memory for execution of the executable application using the reduced amount of memory; and

determining, in response to purging the pre-existing application, whether the client has sufficient available memory for execution of the executable application using the reduced amount of memory.

35. The method of claim 34, further comprising the steps of:

iteratively repeating the purging and determining steps until the client has sufficient available memory for execution of the executable application using the reduced amount of memory;

allocating the reduced amount of memory from the available memory in response to determining that the client has sufficient available memory for execution of the executable application using the reduced amount of memory; and

requesting the executable application from the server upon allocating the reduced amount of memory.

36. The method of claim 34, wherein the step of purging the pre-existing application comprises the step of purging the pre-existing application in accordance with a dynamic list of priorities.

37. The method of claim 28, further comprising the steps of:

compacting a system heap in response to determining that the client has insufficient available memory for execution of the executable application using the reduced amount of memory; and

determining, in response to compacting the system heap, whether the client has sufficient available memory for execution of the executable application using the reduced amount of memory.

38. A communication method comprising the steps of:

establishing a bi-directional communication pathway between a client and a server, the client having a tuner, the client being capable of receiving broadcast information, the server having an executable application, the executable application requiring an amount of memory for proper execution, the server being configured to retransmit data over sequential time intervals;

receiving an input at the client, the input corresponding to the executable application on the server; and

determining, in response to receiving the input, whether the client has sufficient available memory for execution of the executable application.